

1/19

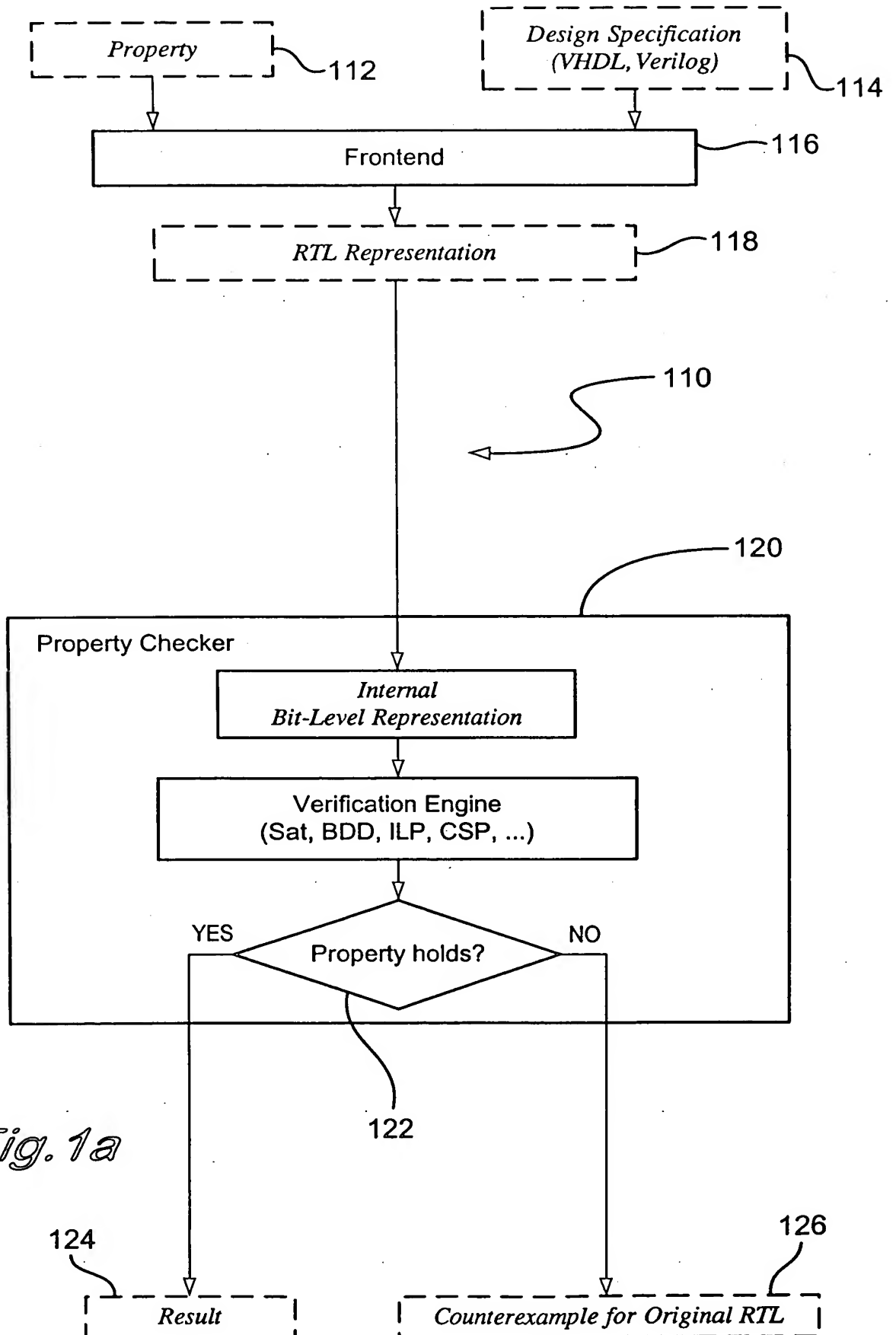


Fig. 1a

2/19

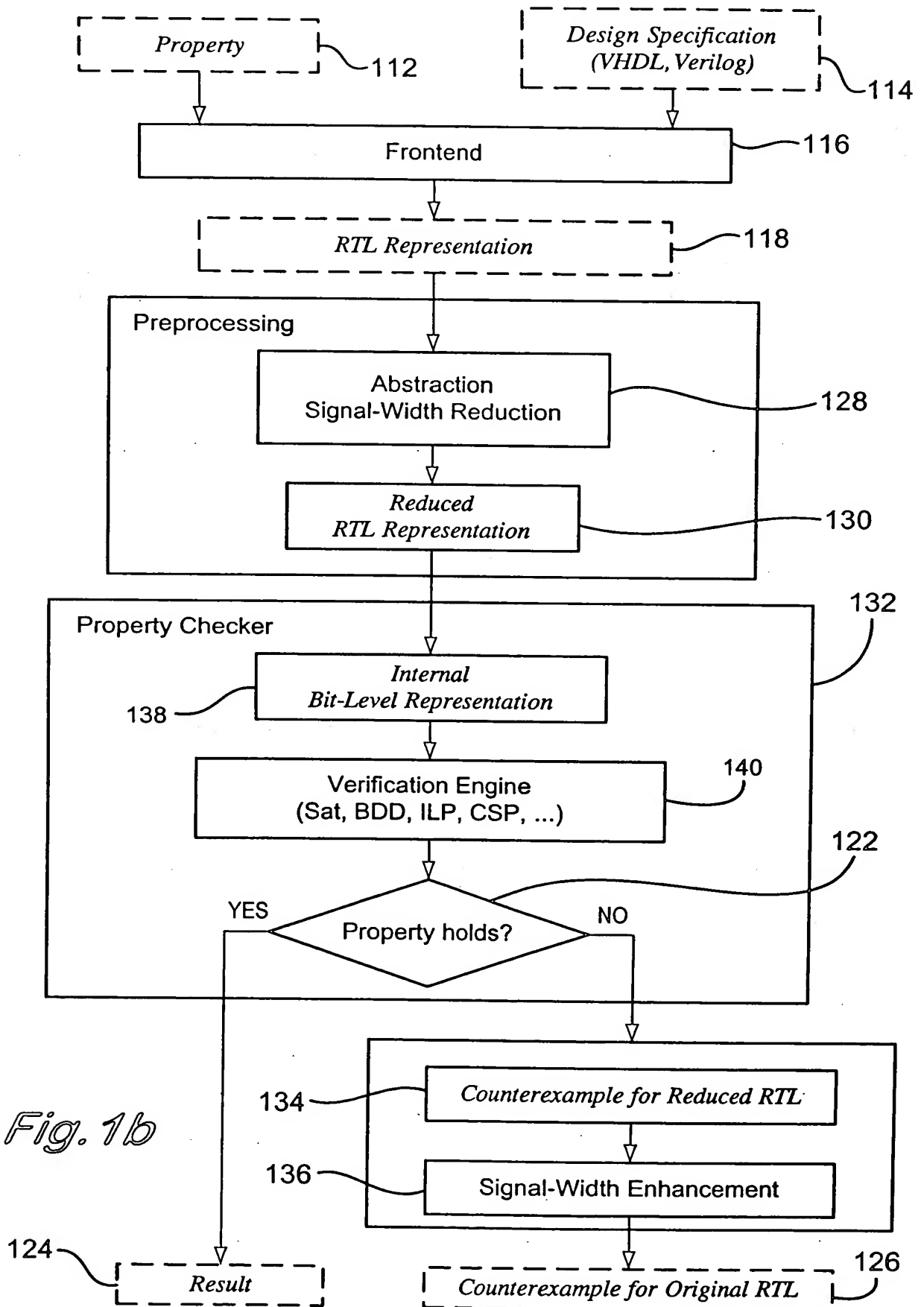


Fig. 1b

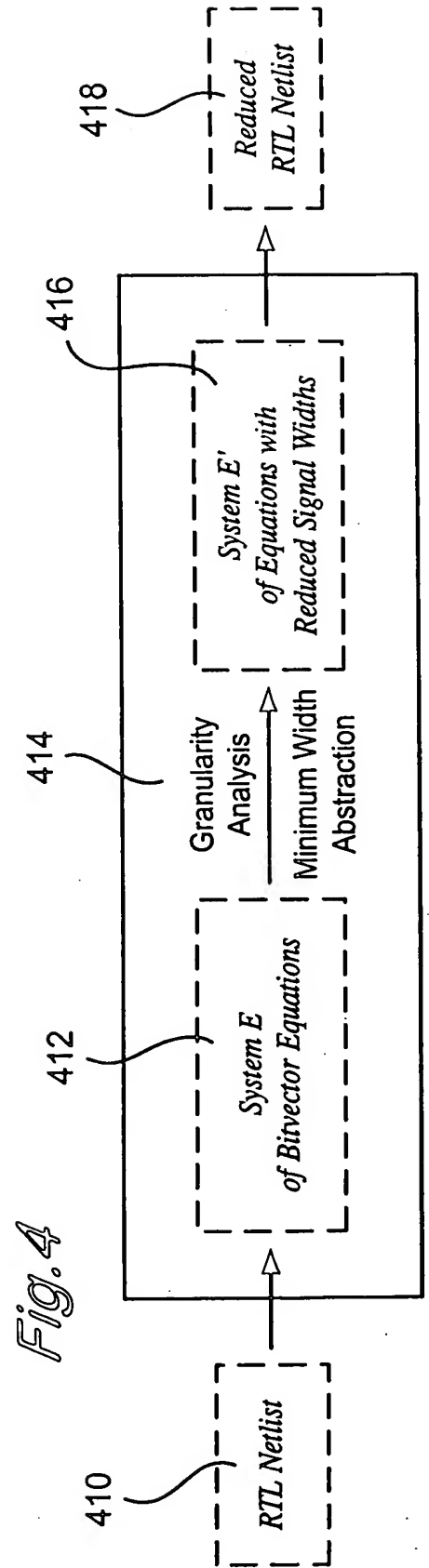
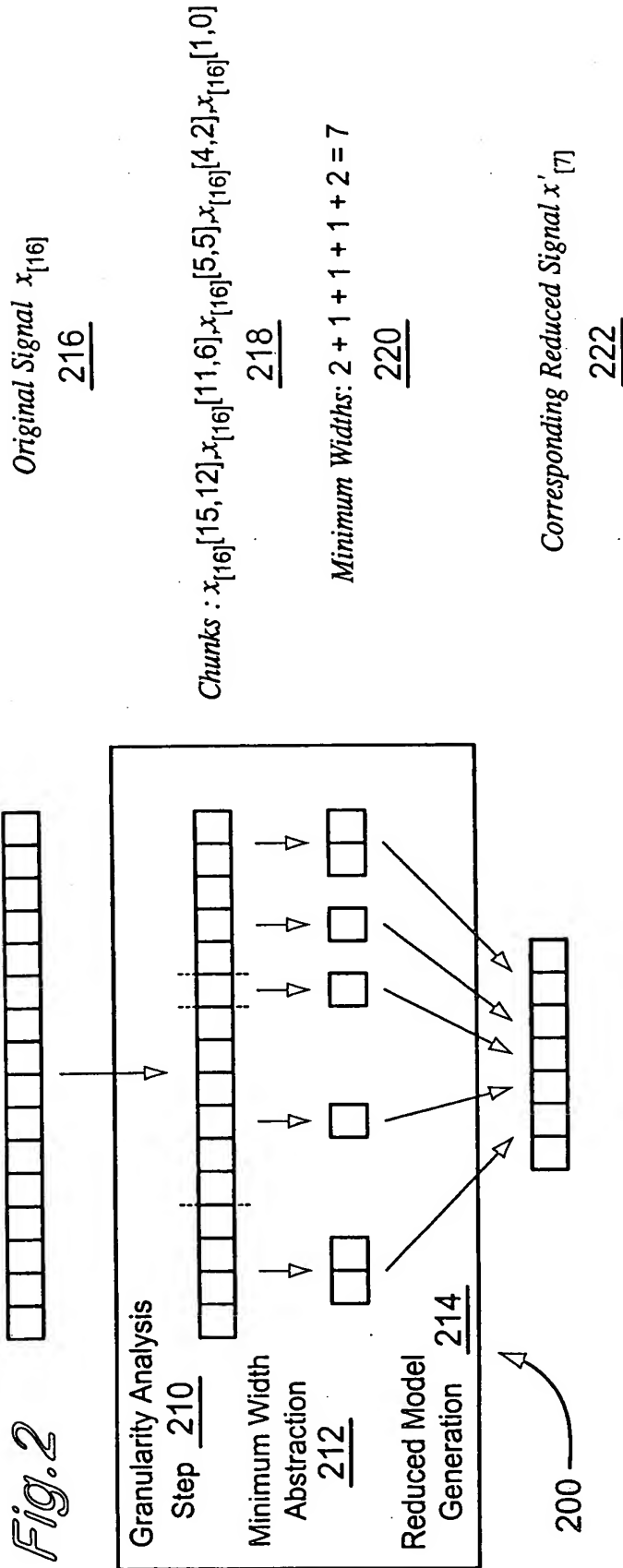


Fig. 3

Bitvector Operator	Syntax	Example
bitvector variables	$x_{[n]}$	$x_{[8]}, y_{[1]}, z_{[4]}, \dots$
bitvector constants	$c_{[m]}$	10011 _[5] , 00111111 _[8] , 0 _[1] , 1 _[1] , ...
concatenation	\otimes	$x_{[16]} \otimes y_{[4]}$
extraction	$[j, i]$	$x_{[8][5, 2]}$
bitwise negation (inversion)		$\text{neg}(x_{[8]})$
bitwise Boolean operations	and, or, xor nand, nor, xnor	$x_{[12]}$ and $y_{[12]}$, $x_{[12]}$ or $y_{[12]}$, $x_{[12]}$ xor $y_{[12]}$ $x_{[12]}$ nand $y_{[12]}$, $x_{[12]}$ nor $y_{[12]}$, $x_{[12]}$ xnor $y_{[12]}$
if-then-else	ite	$\text{ite}(a_{[4]} = b_{[4]}, x_{[8]}, y_{[8]})$ $\text{ite}(a_{[4]} < b_{[4]}, x_{[8]}, y_{[8]})$
arithmetic	+, - *	$x_{[32]} + y_{[32]}$, $x_{[32]} - y_{[32]}$ $x_{[16]} * y_{[16]}$
memory read memory write	$\text{mem}_{[m..n]}^{[i..j]}$	$x_{[10]} := \text{mem}_{[128..10]}^{[i..7]}$ $\text{mem}_{[32..8]}^{[i..5]} := x_{[8]}$

5/19

203010-0288007

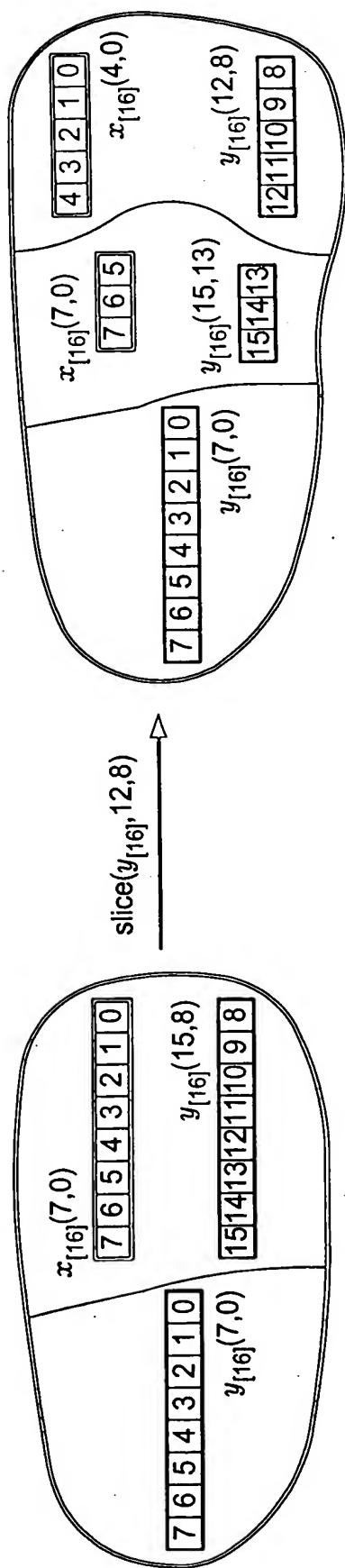


Fig. 5

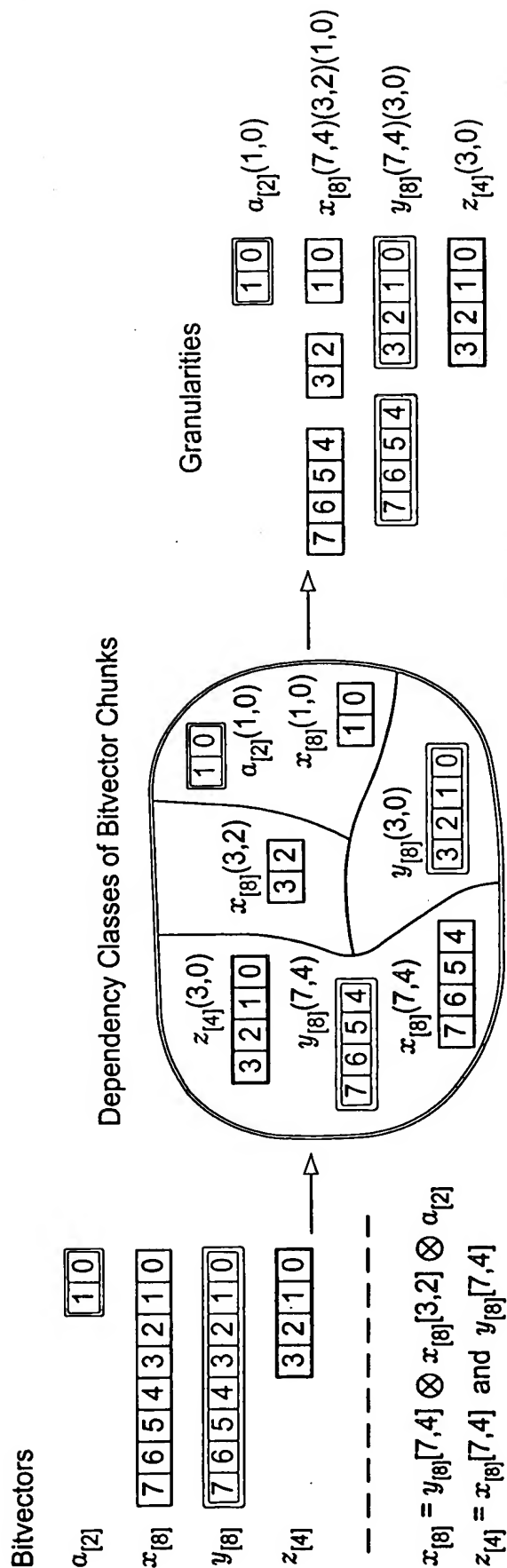


Fig. 6

6/19

Fig. 6a

Process 1 Granularity Analysis of Bitvector Equations

```

1  gran(e) {
2  switch (e);
3  case e ≡ 's[n] = t1[m1] ⊗ t2[m2]':
4  case e ≡ 's[n] = (t1[m1] ⊗ t2[m2])(j,i)':
5
6
7
8
9
10
11 case e ≡ 's[n] = (t[m][l,k])(j,i)':
12 case e ≡ 's[n] = ite(a[m] = b[m], tt[n], te[n])(j,i)':
13 case e ≡ 's[n] = (t1[m] and t2[m])(j,i)':
14 ...
15 case e ≡ 's[n] = neg(t[n])':
16 case e ≡ 's[n] = ite(a[m] = b[m], tt[n], te[n])':
17 case e ≡ 's[n] = t[n] and t2[n]':
18 ...
19 case e ≡ 'x[n](j,i) = y[m](l,k)':
20 }
```

```

gran('s[n][n-1, m2] = t1[m1]'): gran('s[n][m2-1,0] = t2[m2]'):
if (j < m2) {
    gran('s[n] = t2[m2](j,i)'):
} else if (i ≥ m2) {
    gran('s[n] = t1[m1](j-m2, i-m2)'):
} else {
    gran('s[n] = t1[m1](j-m2, 0) ⊗ t2[m2](m2-1, i)'):
}

gran('s[n] = t[m](k+j, k+i)'):
gran('s[n] = ite(a[m] = b[m], tt[n](j,i), te[n](j,i))'):
gran('s[n] = t1[m](j,i) and t2[m](j,i)'):

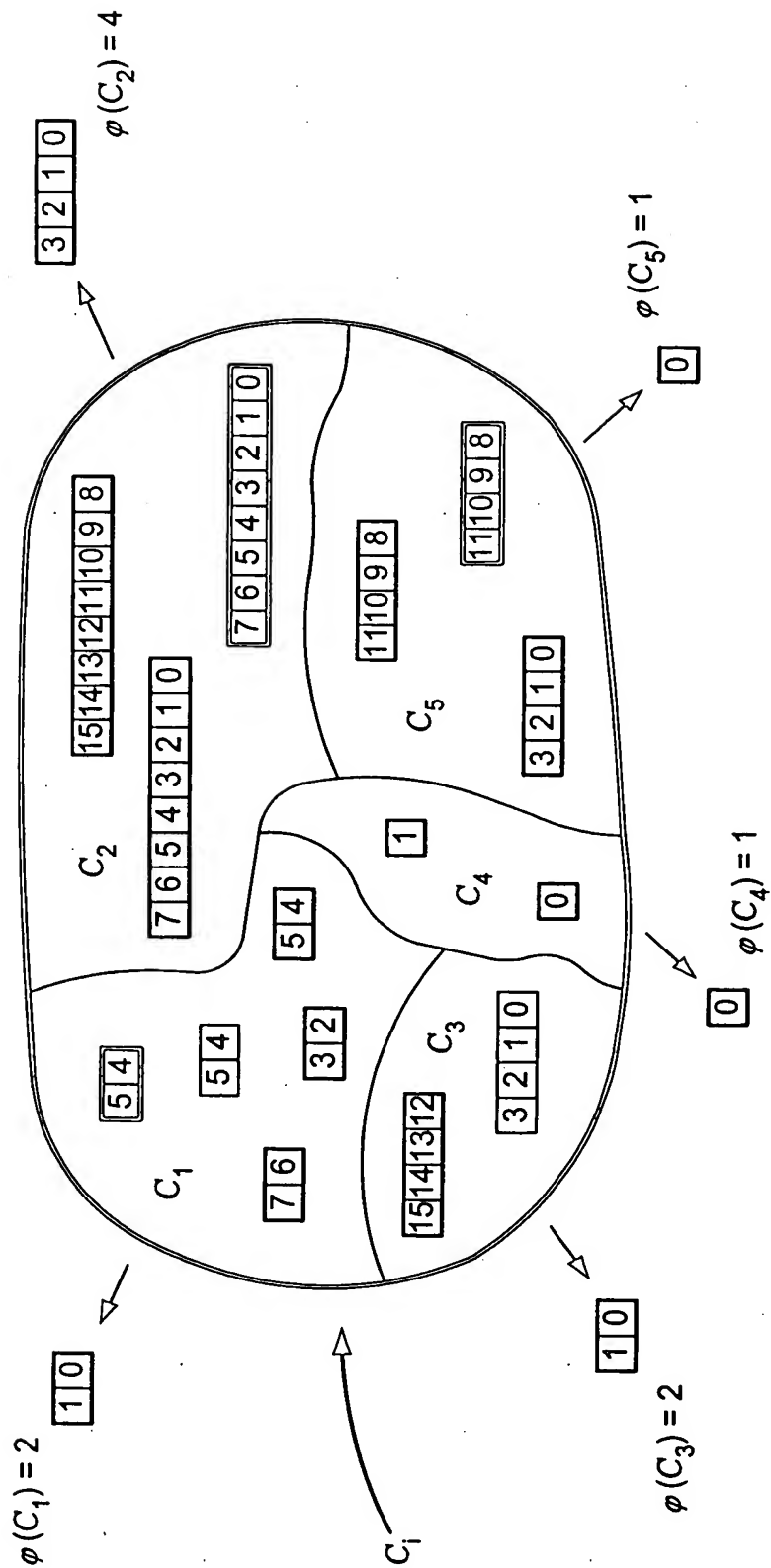
gran('s[n] = t[n]'):
gran('a[m] = b[m]'): gran('s[n] = tt[n]'): gran('s[n] = te[n]'):
gran('s[n] = t1[n]'): gran('s[n] = t2[n]'):

slice(x[n](j,i); slice(y[m](l,k); join(x[n](j,i), y[m](l,k))):

```

7/19

Fig. 7



8/19

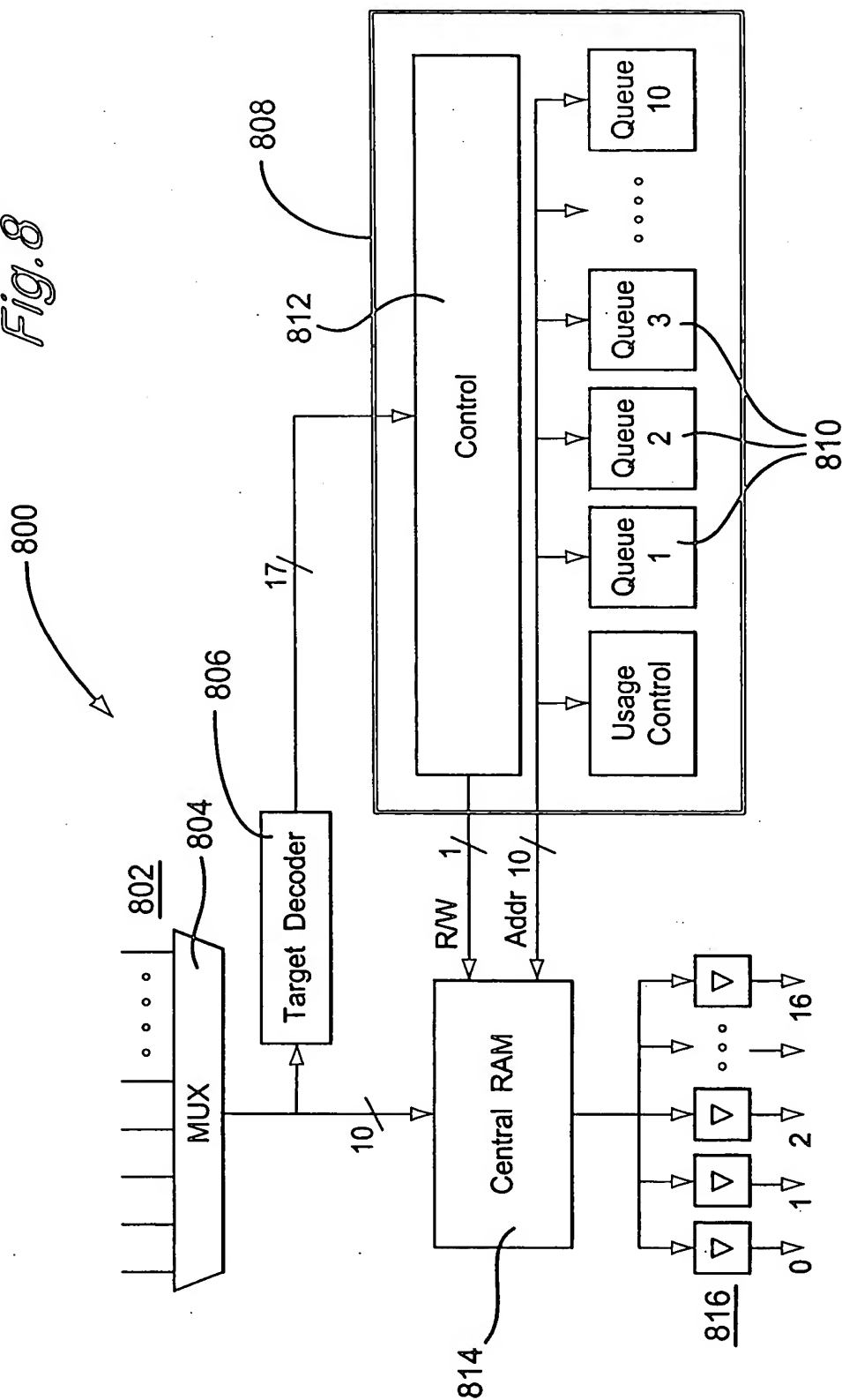
Fig. 7a

Process 2 Reduced Model Generation

```
1  for each bitvector variable  $x_{[n]}$  {  
2     $m := 0$  ;  
3    for each chunk  $x_{[n]} \langle j, i \rangle$  of the computed granularity of  $x_{[n]}$  {  
4       $C := find ( x_{[n]} \langle j, i \rangle )$  ; // equivalence class containing  $x_{[n]} \langle j, i \rangle$   
5       $m := m + \varphi(C)$  ;  
6    }  
7    if  $(m \geq n)$  then  $m := n$  ;  
8    replace all occurrences of  $x_{[n]}$  in the bitvector equations by  $x'_{[m]}$  ;  
9    and adjust all extraction expressions affected by  $x_{[n]}$  ;  
10 }
```

9/19

Fig. 8



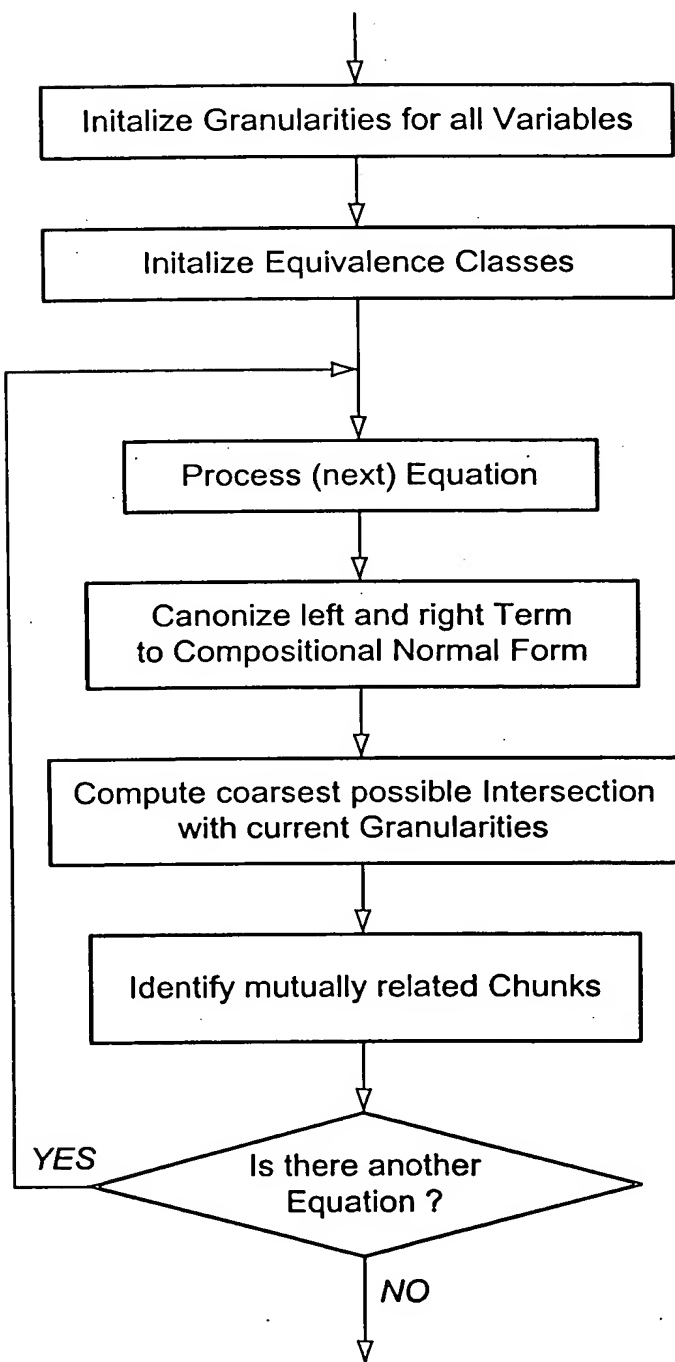
10/19

208010" 0488E001

Fig.9

	Property	Original design	Reduced model
Computation times of the prototype for analysis and reduced model generation	nop read write		2.96 secs 6.53 secs 3.24 secs
FIFO sizes on RTL	nop read write	160 cells x 10 bit 160 cells x 10 bit 160 cells x 10 bit	160 cells x 2 bit 160 cells x 3 bit 160 cells x 3 bit
Overall number of bits in all relevant signals (cones of influence of the property)	nop read write	20925 31452 14622	5034 (24.0 %) 10592 (33.6 %) 5163 (35.3%)
Overall number of gates in synthesized netlist	nop read write	23801 23801 23801	5661 (27.9 %) 7929 (33.3 %) 7929 (33.3 %)
Number of state bits	nop read write	1658 1658 1658	362 (21.8 %) 524 (31.6 %) 524 (31.6 %)
Property checker runtimes	nop read write_fail write_hold	23:33 min 42:23 min 2:08 min 27:08 min	37.96 secs (2.7 %) 3:27 min (8.1 %) 25.66 secs (19.5 %) 1:08 min (4.2 %)

11/19



$$x_{[8]}[7,2] \otimes x_{[8]}[1,0] = (a_{[4]} \otimes b_{[4]} \otimes c_{[4]})[11,4]$$

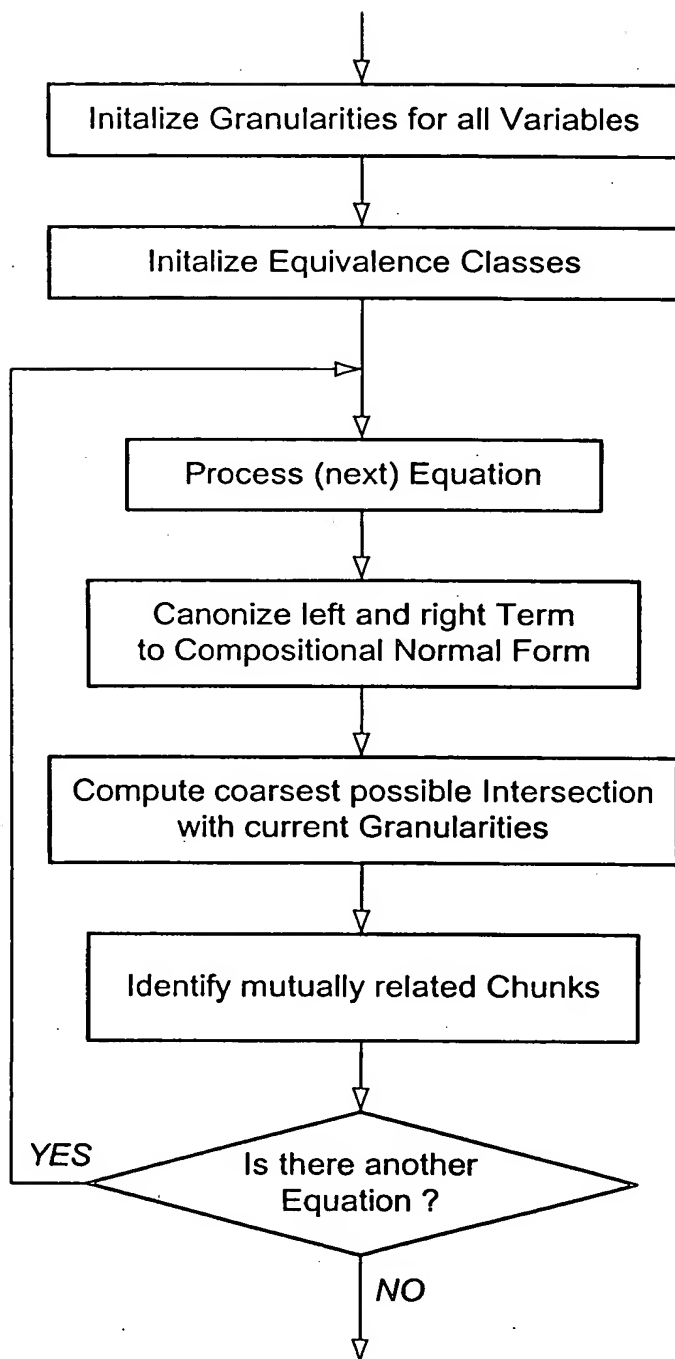
$$x_{[8]}[7,0] = a_{[4]}[3,0] \otimes b_{[4]}[3,0]$$

$$x_{[8]}[7,4] \otimes x_{[8]}[3,0] = (a_{[4]}[3,0] \otimes b_{[4]}[3,0])$$

$$\begin{aligned} x_{[8]}[7,4] &\longleftrightarrow a_{[4]}[3,0] \\ x_{[8]}[3,0] &\longleftrightarrow b_{[4]}[3,0] \end{aligned}$$

Fig. 10

12/19



$$x_{[8]} = \text{ite}(a_{[4]} = b_{[4]}, y_{[8]}, z_{[8]})$$

$$x_{[8]} = \text{ite}(a_{[4]} = b_{[4]}, y_{[8]}, z_{[8]})$$

$$x_{[8]}[7,0] = \text{ite}(a_{[4]}[3,0] = b_{[4]}[3,0], y_{[8]}[7,0], z_{[4]}[7,0])$$

$$a_{[4]}[3,0] \longleftrightarrow b_{[4]}[3,0]$$

$$x_{[8]}[7,0] \longleftrightarrow y_{[8]}[7,0]$$

$$x_{[4]}[7,0] \longleftrightarrow z_{[8]}[7,0]$$

Fig. 11

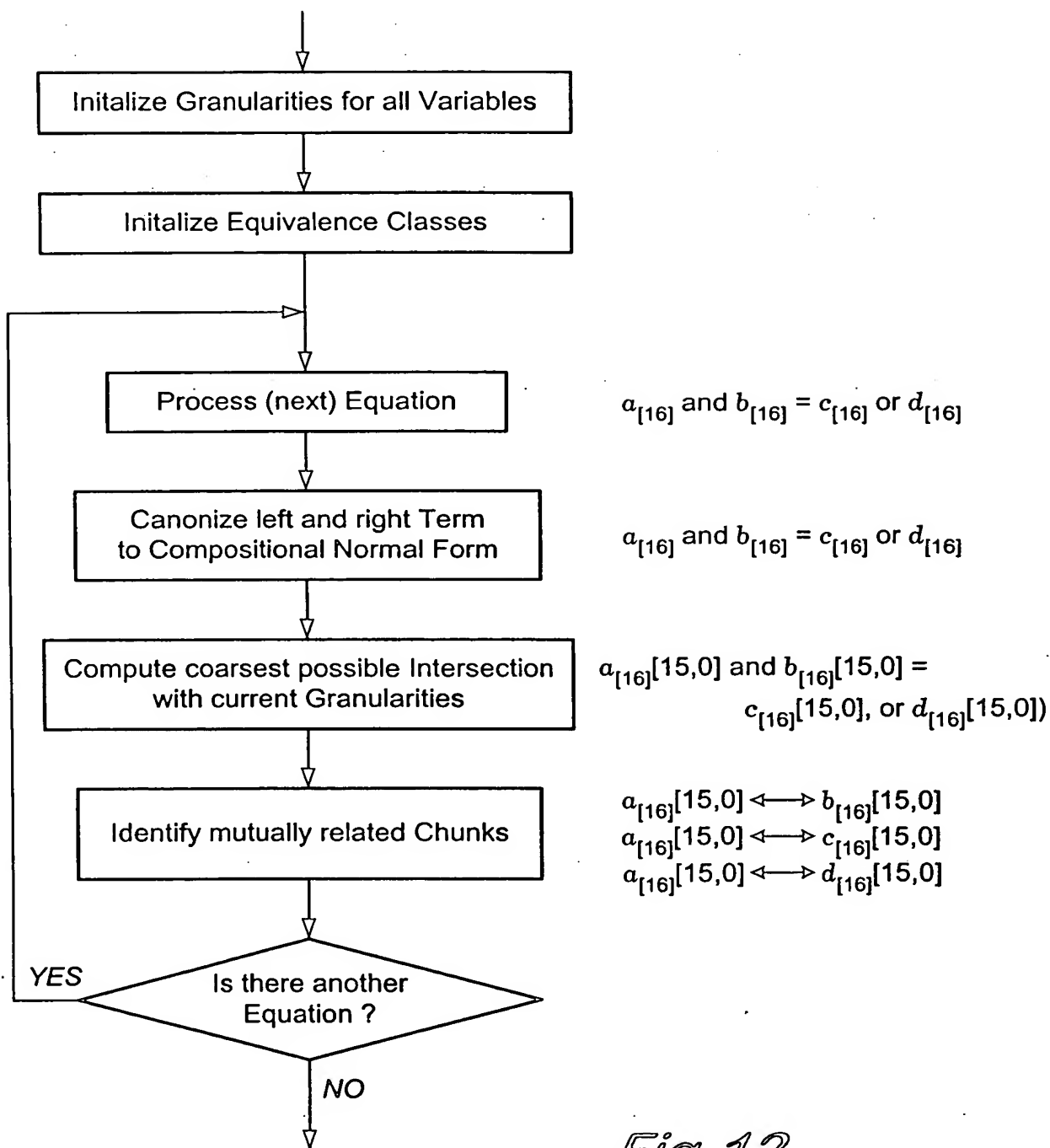


Fig. 12

14/19

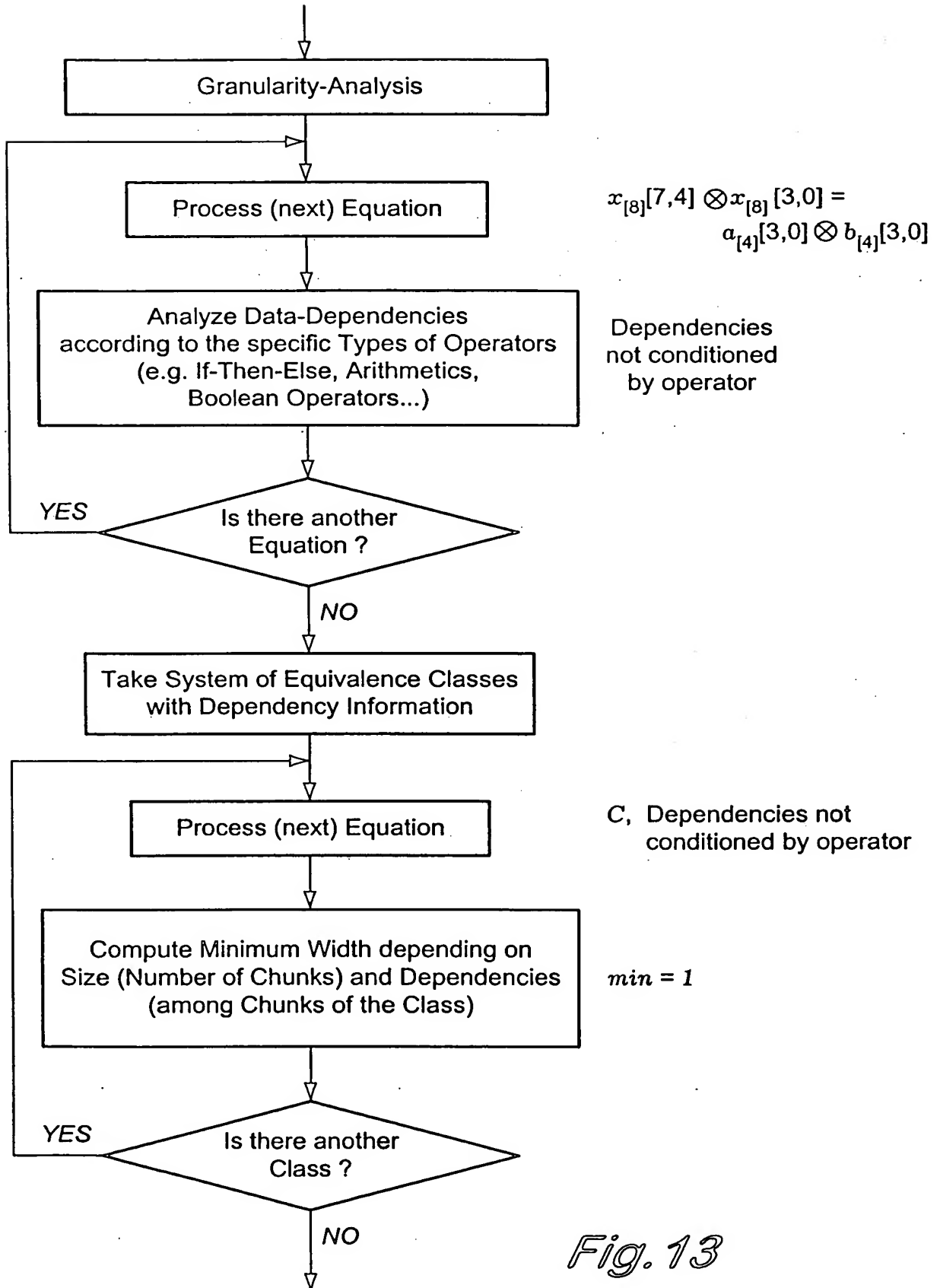


Fig. 13

15/19

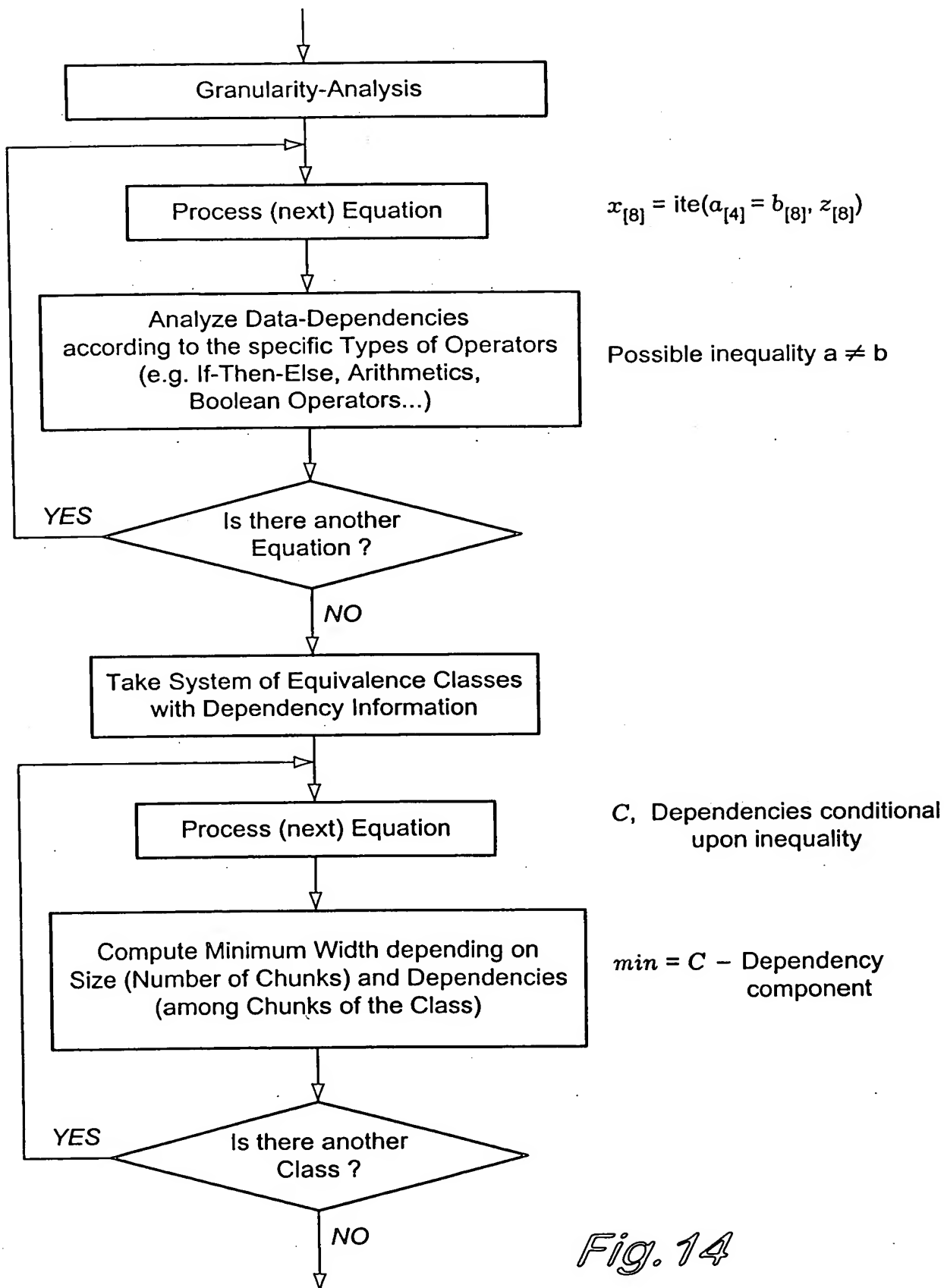


Fig. 14

10038670.010602

16/19

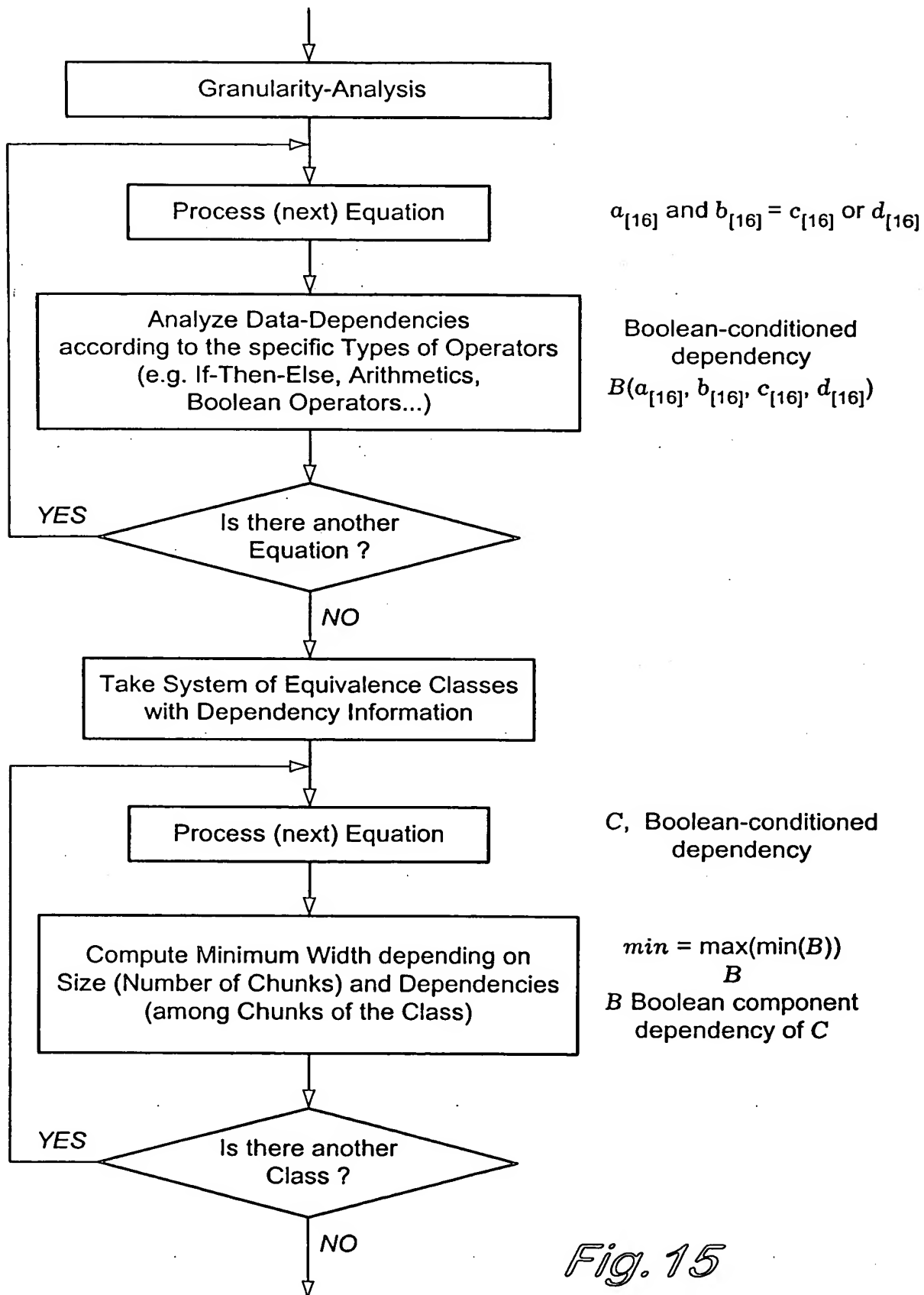


Fig. 15

17/19

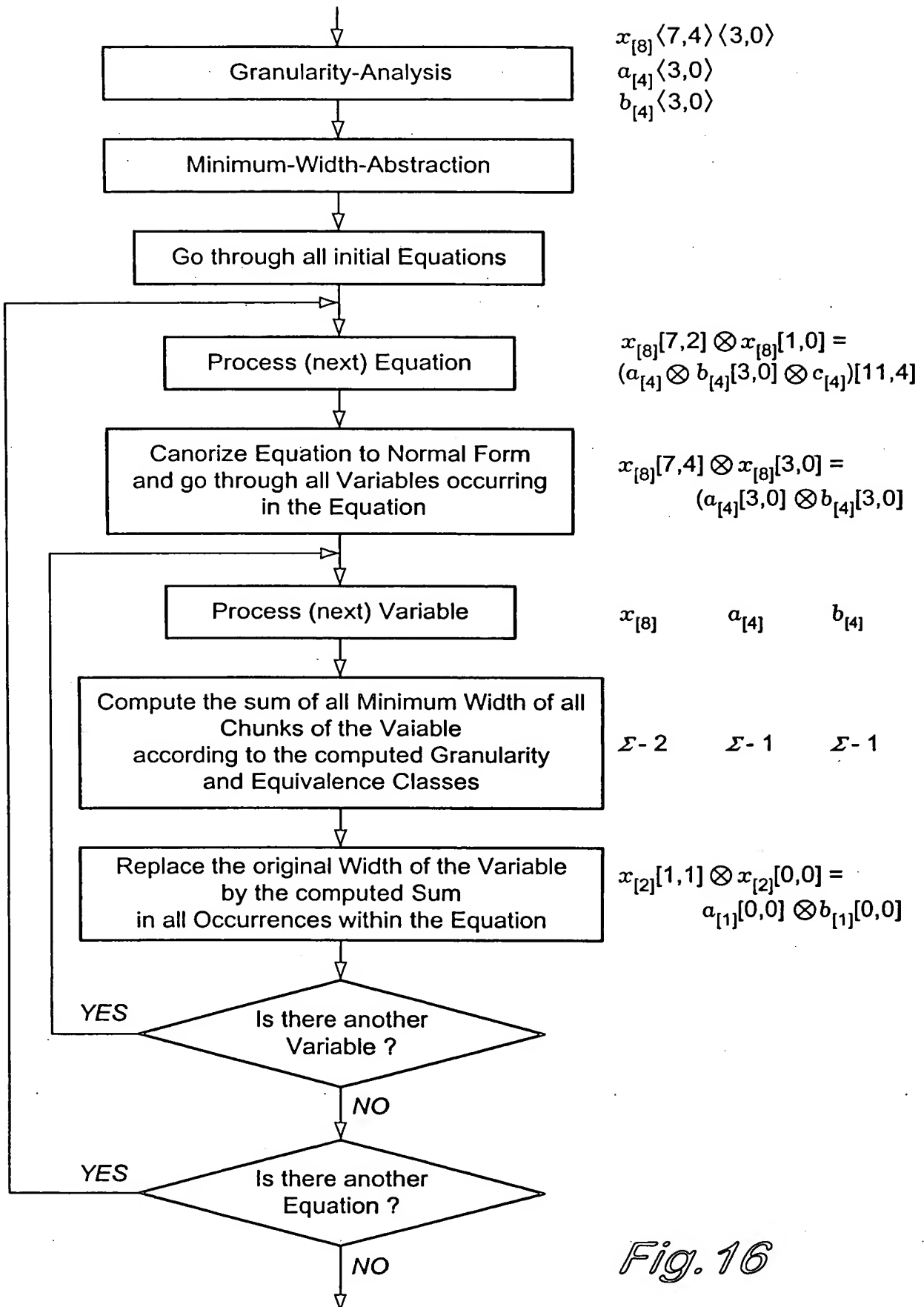
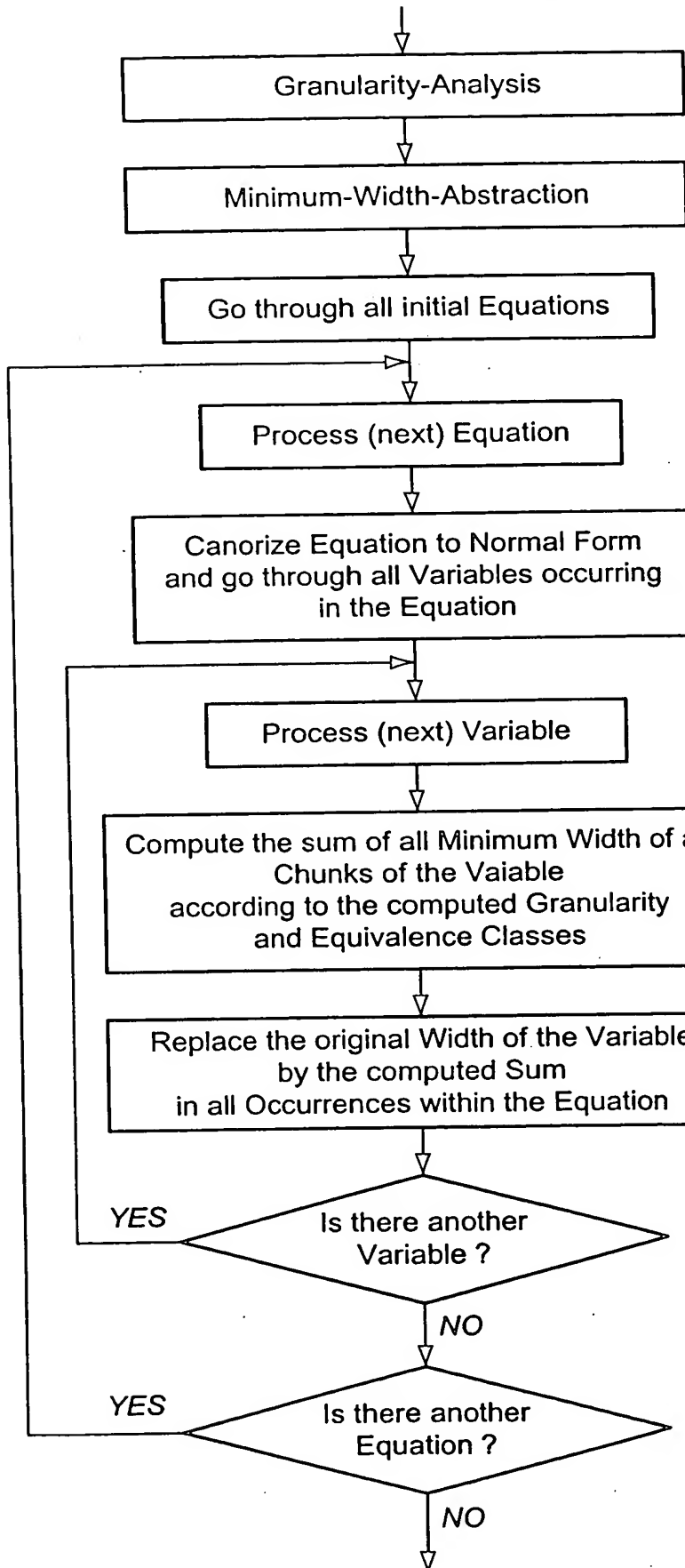


Fig. 16

18/19



$x_{[8]} \langle 7,6 \rangle \langle 5,0 \rangle$
 $y_{[8]} \langle 7,6 \rangle \langle 5,0 \rangle$
 $x_{[8]} \langle 7,4 \rangle \langle 3,0 \rangle$
 $a_{[4]} \langle 3,0 \rangle$
 $b_{[4]} \langle 3,0 \rangle$

$$x_{[8]} = \text{ite}(a_{[4]} = b_{[4]}, y_{[8]}, z_{[8]})$$

$x_{[8]} \quad y_{[4]} \quad z_{[8]} \quad a_{[4]} \quad b_{[4]}$

$$\begin{array}{ccc} \Sigma = 3 & \Sigma = 3 & \Sigma = 3 \\ \Sigma = 1 & \Sigma = 1 & \end{array}$$

$$x_{[3]} = \text{ite}(a_{[1]} = b_{[1]}, y_{[3]}, z_{[3]})$$

Fig. 17

19/19

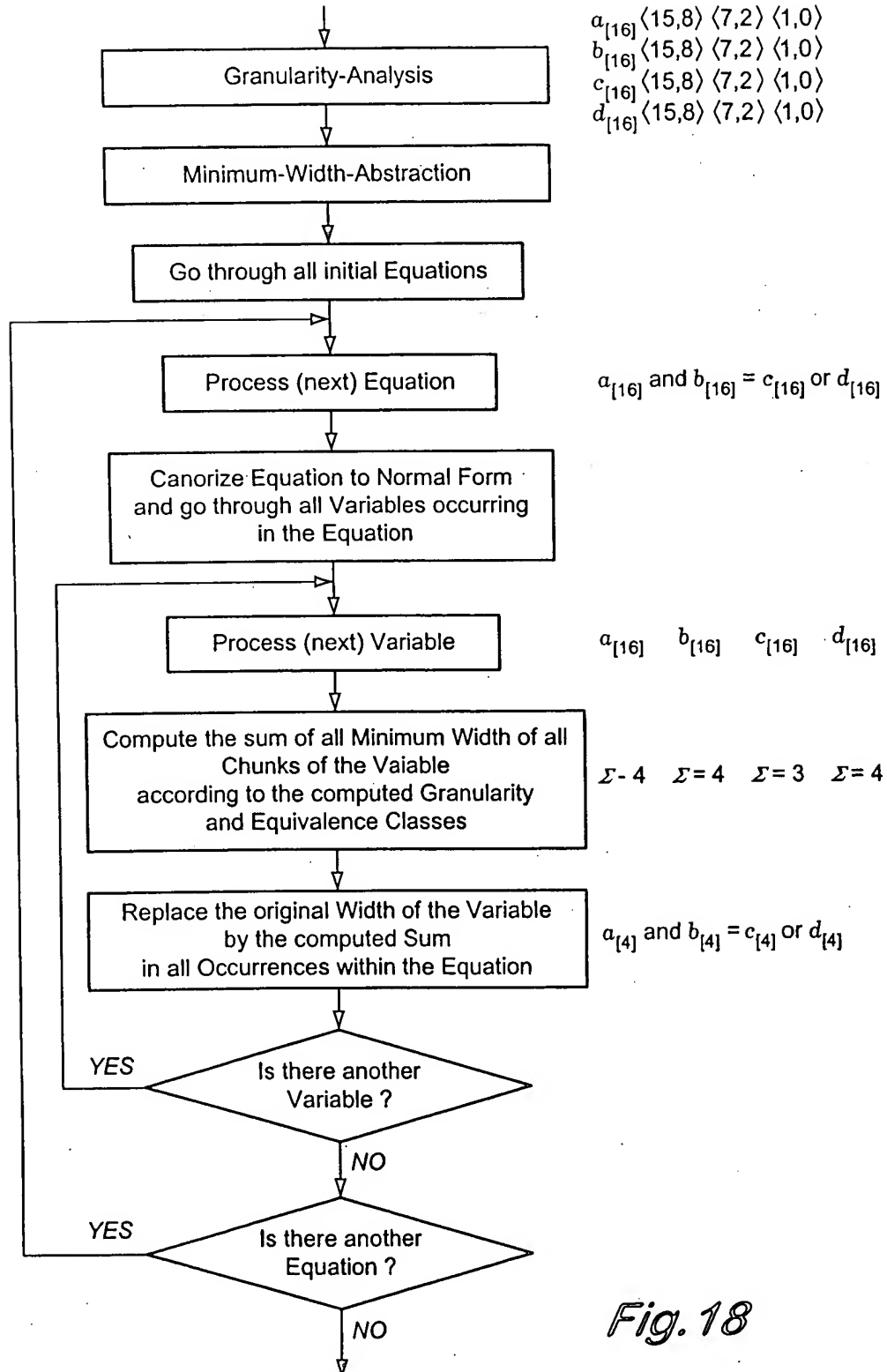


Fig. 18